CMR Workload Specification

Overview

This document is intended to guide load testing of the Common Metadata Repository (CMR). It is intended to describe starting conditions, testing parameters, and reporting requirements for these tests, but is not intended to prescribe implementation or technology strategies.

Defining Sub-Second Search

The CMR is being designed to provide sub-second searching, and as such the definition of sub-second search should be clarified. Based on initial prototype investigations outline the qualities of sub-second search:

Sub-second search implies that average search time is < 1s for both collection and granule level searches as measured from the time the CMR receives a query until it has completed streaming the response.

Sub-second performance will be provided for queries that have the following:

- *Granule queries that include a collection identifier such that it identifies a single collection. That could be ECHO Collection Id or provider id with dataset id or short name and version id.*
- *Number of results requested is 100 or fewer.*
- *Requested format is either a strict reference or the format given by the provider or the format has been pre-computed and cached.*
- *Any of the granule conditions including temporal and spatial will be supported with sub-second search.*
- *Any number of conditions will be supported.*

Environmental Considerations

From an environmental viewpoint, both the hardware and software employed in the workload need to reflect the operational buildout. This means that the following items need to be kept in sync between systems:

- **Hardware (or VM) counts and specifications:** database nodes, elastic nodes, etc.
- **Network configuration/capacity/security**
- **OS Versions and Patching Levels**
- **Software Platform and Versions:** applications containers and runtime environments such as the JVM and component application servers
- **3rd Party Library Versions and Patching:** any libraries and software packages (e.g. Oracle, ElasticSearch, Quartz, etc) used by the CMR

Ideally, both environments would be built via a configuration management tool such as Puppet or Chef. If there are any discrepancies among these items, any reports resulting from workload tests need to document these discrepancies.

Search and Retrieval Testing Criteria

This section outlines the expectations of the CMR under various search loads. As stated earlier, the CMR is expected to sustain sub-second average search times under normal system load and continue to provide acceptable performance under a stress test. The tests outlined here will

simulate these load environments and establish criteria for determining success. There are two tests included in this document:

- Sustained Load Testing (AKA 24 Hour Load Test)
- Spike Load Testing

Initial System State

The system should be preloaded with a mirror of the operational search index and database prior to the execution of the test. This mirror will be periodically updated as described in the "Frequency of Refresh".

Search Load Characteristics

Types of Searches

Load on the system should represent actual system load, meaning that query parameters should mimic searches performed on the production system. The workload system should use actual operational queries to generate search requests.
This load includes a mix of searches to include various combinations of spatial, temporal and keyword searches at both the collection and granule level as well as direct collection and granule retrieval using concept-ids. In addition, queries may be returned in various supported formats (e.g. json, ECHO10, ISO 19115, DIF)

Concurrent Requests

The number of concurrent requests in the system should be derived from load characteristics and should be configurable at runtime. More details on appropriate settings are explained in the specific sections below addressing sustained and spike load tests.

ACLs and Load

For the purposes of these tests it is to be assumed that the token(s) used to generate load have the same ACLs as a normal registered URS user accessing the CMR. Administrators and non-authenticated guest users should not be used for generating search load. Future versions of the workload system should work to accommodate a more representative mix of user and access levels.

Sustained Load Testing

This test involves executing a predetermined number of queries on the system and reporting statistics based on those queries. Sustained load should reflect 125% of the most active search day over the past three months or the highest target established since the outset of load testing. This load target can be determined via log queries and should be updated quarterly as needed.

Concurrency

The test should run through all of the queries and report metrics as outlined below. The number of concurrent users should be determined as shown in the following example.

Example

If the most active single search day in past three months reports 300,000 queries, the sustained load test should issue 375,000 (1.25 * 300,000) queries. To determine the number of concurrent users, assuming constant load, look at the average number of queries per minute, 260 (375,000/24 hours/60 minutes) and assuming an average response time of 1 sec, the system should require 4 (260/60) requests each second. This gives a concurrent user count of 4 using this rough

approximation.
More succinctly, the following formula should be used:
Concurrent users = (B * 1.25) / 24h / 60m / 60s

Establishing a Baseline for Target Load

Because the CMR is evolving from ECHO and the GCMD, the initial load baseline should be based on an aggregate of current sustained load on each system as determined by available metrics.

Spike Load Testing

(Note: This has not been implemented in CMR at this point, only sustained load tests are currently supported)

If the intention of the Sustained Load Test is to ensure sub-second performance, the intention of Spike Load Testing is to push that performance to its limits and observe system performance under times of extreme stress.

Load Ramp Up

This test will begin with a small number of concurrent workers running at sustained load as described above. The number of concurrent users will then be periodically increased over set intervals. These the number and spacing of timing intervals should be configurable, as should the initial number of concurrent users.

Example

(Note: These data are not based on any actual metrics gathered via CMR load testing an are randomized numbers being used to illustrate principles of the spike load test.)
Figure 1 shows an example ramp up, starting with 4 users for the first 10 minutes of the test and then doubling in number for each successive 10 minute interval. This data is collected over 6 intervals. The next figure, 2 depicts system response time overlaid on the concurrent users line. This example data shows that system performance degrades significantly as the number of users increases beyond 32 concurrent users.

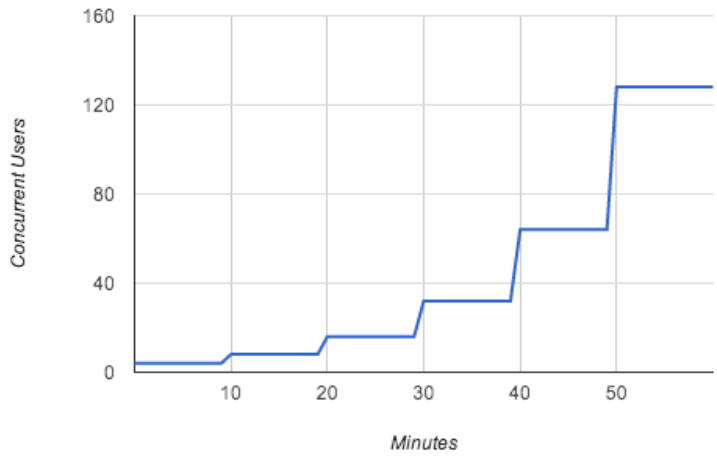

Figure 1 : Example Spike Load Test Ramp Up

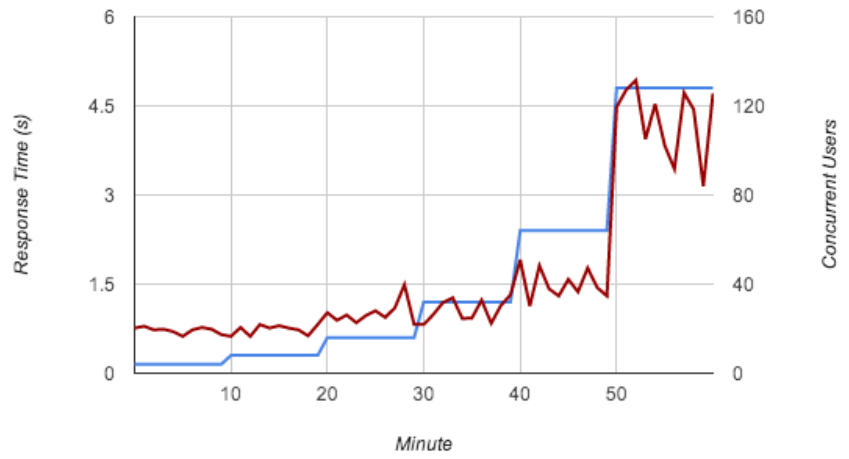

Figure 2 : Example Spike Load Test Performance Chart

Post-Test Reporting and Pass/Fail Criteria

Because the CMR is expected to perform sub-second searches under sustained load, reports generated from workload runs should provide a clear status at the top to indicate whether sub-second average query time is achieved over the duration of the load test.
In addition, the following summary metrics should be reported after each Search and Retrieval test run:

- Total Time of Test Duration
- Total Number of Concurrent Users Executing Queries
- Query Count Breakdown Statistics
    - Overall Query Count
    - Collection Query Count
    - Granule Query Count
    - Overall Spatial Query Count: This includes any successful query that contained a spatial component.
    - Collection Spatial Query Count: This includes any successful collection level query that contained a spatial component.
    - Granule Spatial Query Count: This includes any successful granule level query that contained a spatial component.
    - Single Granule Retrieval Count: This includes any successful query that is retrieving a single granule with a given concept-id
    - Count Breakdown by Status Code: This will reveal any unsuccessful queries that result in a non-2xx response code.

Each of the following metrics should be provided for each Query Category in the Breakdown list above:

- Minimum query time
- Maximum query time
- Average query time
- Query time standard deviation

Both Sustained and Spike Load tests should report these metrics, however the spike load test should provide these data for each interval, also including the length and number of the interval.

Example Workload Report:

(Note: this report is taken from an actual workload run conducts in August 2014)

Notes:

- The only difference between this and the previous run is that I fixed the platform, instrument, sensor search bug that caused poor performance.
- 86 granule queries took longer than 1s. (out of 90K granule queries)
  - 79 of those were retrieving ECHO10 metadata from Oracle.
- Average granule performance is *under 100 ms* now with a standard deviation of 130 ms.

Granules

Total requests - 89154

Min - 6

Max - 4080

Average - 74

Standard Deviation - 130.78

Collections

Total requests - 274440

Min - 6

Max - 5020

Average - 31

Standard Deviation - 62.69

Granule Search Format Breakdown

CSV

query-count: 371

min: 19

max: 93

avg: 26

std-dev: 12

XML

query-count: 1855

min: 19

max: 105

avg: 26

std-dev: 12

JSON

query-count: 8908

min: 21

max: 767

avg: 32

std-dev: 21

ATOM

query-count: 17449

min: 16

max: 4080

avg: 100

std-dev: 119

ECHO10

query-count: 60571

min: 6

max: 2193

avg: 75

std-dev: 142

Granule Search Condition Breakdown

parameters: :instrument :online_only :echo_collection_id :temporal :platform :sensor

query-count: 1113

min: 45

max: 1000

avg: 102

std-dev: 53

parameters: :dataset_id :echo_collection_id :bounding_box

query-count: 742

min: 19

max: 103

avg: 26

std-dev: 12

parameters: :online_only :echo_collection_id

query-count: 371

min: 17

max: 81

avg: 22

std-dev: 10

parameters: :producer_granule_id :echo_collection_id

query-count: 372

min: 22

max: 122

avg: 30

std-dev: 14

parameters: :short_name :version :provider :temporal

query-count: 2228

min: 30

max: 2099

avg: 41

std-dev: 52

parameters: :online_only :echo_collection_id :bounding_box

query-count: 1113

min: 19

max: 1743

avg: 89

std-dev: 102

parameters: :echo_collection_id

query-count: 1860

min: 16

max: 214

avg: 49

std-dev: 30

parameters: :online_only :dataset_id :echo_collection_id :bounding_box :temporal

query-count: 1484

min: 21

max: 105

avg: 27

std-dev: 11

parameters: :short_name :version :provider

query-count: 742

min: 55

max: 237

avg: 66

std-dev: 17

parameters: :online_only :echo_collection_id :temporal

```
  query-count: 371
  min: 32
  max: 117
  avg: 40
  std-dev: 12

parameters: :short_name :bounding_box :version :provider
  query-count: 1113
  min: 431
  max: 4080
  avg: 483
  std-dev: 114

parameters: :echo_granule_id
  query-count: 74676
  min: 6
  max: 2193
  avg: 69
  std-dev: 130

parameters: :online_only :echo_collection_id :bounding_box :temporal
  query-count: 2969
  min: 52
  max: 1302
  avg: 124
  std-dev: 66
```

Table 1 - Sample Report for Sustained Load Test

Frequency of Baseline Refresh

Once the baseline performance for each criteria outlined in this document has been established, it is expected that the environment used to execute search load testing is refreshed and mirrored from the operational system at least 4 times annually. This ensures that the testing is kept in step with production with respect to inventory size, system load, as well as hardware and software versions.

Ingest Load and Index Criteria

This document will be updated at a later time to reflect workload specification for ingest and indexing performance. ECHO's current ingest/index load test specifications should should be adapted to the CMR as part of a later phase.

Ordering/Services Criteria

This document will be updated at a later time to reflect workload specification for ordering and service invocation performance.

References

- Performance Testing Guidance for Web Applications {+}http://msdn.microsoft.com/en-us/library/bb924375.aspx+Load Testing Calculators {+}http://www.webperformance.com/library/tutorials/CalculateNumberOfLoadtestUsers/+